COMPOSITE DESIGN PATTERN

TechTrio - Mikka Ella, Ashley Moriah, Herminigildo

What is composite pattern?

- is a partitioning design pattern and describes a group of objects that is treated the same way as a single instance of the same type of object?
- should be used when clients need to ignore the difference between compositions of objects and individual objects.

History

- This pattern was first introduced by the "Gang of Four" (Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides) in their book "Design Patterns: Elements of Reusable Object-Oriented Software," which was published in 1994.

Pros

- You can work with complex tree structures more conveniently: use polymorphism and recursion to your advantage.
- Open/Closed Principle. You can introduce new element types into the app without breaking the existing code, which now works with the object tree.

Cons

- It might be difficult to provide a common interface for classes whose functionality differs too much. In certain scenarios, you'd need to overgeneralize the component interface, making it harder to comprehend.

Real Life Examples

- Company
- Menu Systems

Four participants

- Component - declares the interface for objects in the composition and for accessing and managing its child components.
- Leaf - defines behavior for primitive objects in the composition.

- Composite - stores child components and implements child related operations in the component interface.
- Client - manipulates the objects in the composition through the component interface.

References:

- Composite Pattern - Javatpoint
- Composite (refactoring.guru)
- Composite Design Pattern - GeeksforGeeks
- Composite Design Pattern. The Composite design pattern is used to… | by raj | Medium